บทความนี้จะบอกคุณในสิ่งที่คุณอยากรู้เกี่ยวกับ IBM คีย์บอร์ด และคุณจะสามารถ "เล่น" กับมันได้อย่างไรบ้าง เช่น กด ctrl-alt-del แล้วคอมพิวเตอร์ไม่ reboot หรือเปลี่ยนสลับที่คีย์บนแป้นใหม่ และแม้กระทั่ง(สำหรับคุณที่มีภาษาไทย) เปลี่ยนคีย์บอร์ดจากแป้น "เกษมณี" มาเป็นแป้น "ปัตตโชติ" หรือแป้นอื่นๆตามใจคุณ

มารู้จัก IBM® คีย์บอร์ดกันเถอะ

สุรพล พฤกษ์ไพบูลย์

มารู้จัก IBM คีย์บอร์ดกันเถอะ ครับ! ก็แป้นที่ทั้งคุณกับผมกดกันอยู่ทุกวัน ๆ นั่นแหละ คีย์บอร์ดเป็นสิ่งที่เราคุ้นเคยกันดี เพราะคีย์บอร์ดเป็นอุปกรณ์พื้นฐานอันหนึ่งที่เราใช้ติดต่อกับคอมพิวเตอร์ ซึ่งผมก็จะไม่สาธยายให้ยืดยาวเป็นที่รำคาญล่ะนะครับ ว่ารูปร่างภายนอกของมันมีส่วนสัดเท่าไหร่ และประกอบด้วยส่วนต่าง ๆ อะไรบ้าง แต่ก็หวังว่าเราคงจะพูดถึงสิ่ง ๆ เดียวกัน คือสิ่งที่อยู่ตรงหน้าของคุณ และมีปุ่มให้คุณกดเล่นอยู่ 83 ปุ่มนั่นแหละครับ

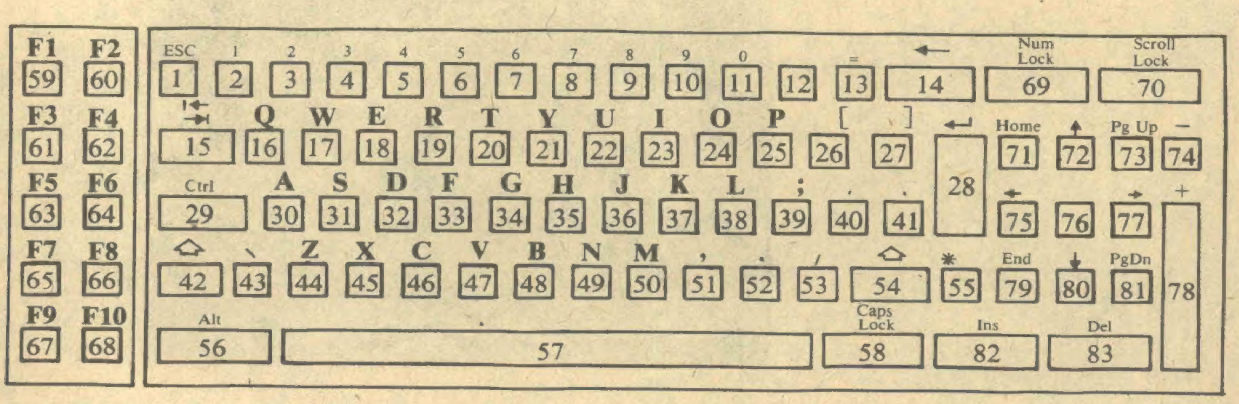
ในตัวคีย์บอร์ดของไอบีเอ็มพีซี จะมีไมโครโปรเซสเซอร์เบอร์ 8048 ซึ่งทำหน้าที่มากมายหลายหลากประการในการควบคุมคีย์บอร์ด แต่หน้าที่หลักอันหนึ่งก็คือ เป็นเสมือนยามคอยสอดส่องดูแลลูกน้องทั้ง 83 คน ว่าคนไหนล่ะครับที่โดนนิ้วของคุณกดอยู่ในช่วงขณะหนึ่ง ๆ ซึ่งเป็นที่แน่นอนล่ะครับว่าทุก ๆ อิริยาบถของคุณกับปุ่มคีย์บอร์ดเล็ก ๆ เหล่านี้ ไม่ว่าคุณจะกดมันเพียงปุ่ม ๆ เดียวหรือพร้อมกันหลายปุ่ม เช่น ctrl-q หรือ alt-k หรือ alt-ctrl-k+p+l+x+y ฯลฯ ก็ตาม เจ้ายาม 8048 ตัวนี้ก็จะรายงานส่งไปที่รอมไบออส (ROM BIOS) ทุกครั้ง และถ้าคุณกดปุ่มใดบนแป้นนานเกินกว่าครึ่งวินาที 8048 ก็จะส่งสัญญาณให้เพิ่มตัวซ้ำเข้าไปในคีย์บอร์ดบัฟเฟอร์ ซึ่งเป็นที่ ๆ ใช้เก็บตัวอักษรที่ถูกพิมพ์ผ่านเข้าไปในเครื่องทันที

ทุกครั้งที่คุณกดคีย์

ทุก ๆ ครั้งที่เรากดคีย์บนแป้น มีอะไรเกิดขึ้นบ้างก่อนที่จะปรากฏเป็นตัวอักษรให้เราเห็นบนจอภาพ อย่างแรกวงจรในคีย์บอร์ดจะส่งสัญญาณเป็นข้อมูลที่มีขนาด 1 ไบท์ ข้อมูลนี้คือค่าสแกนโค้ด (scan code) ซึ่งแต่ละคีย์จะมีค่าแตกต่างกันตั้งแต่ 1 ถึง 83 เช่นคีย์ "Q" มีค่า สแกนโค้ดเท่ากับ 16 คีย์ "A" มีค่าสแกนโค้ดเท่ากับ 30 เป็นต้น (ดูรูปที่ 1 ประกอบ) และเมื่อเรายกนิ้วขึ้นจากคีย์เดียวกันนี้ สแกนโค้ดอีกค่าหนึ่งจะถูกส่งออกมาจากวงจรของคีย์บอร์ด ซึ่งค่าสแกนโค้ดครั้งนี้จะมีค่าเท่ากับค่าสแกนโค้ดของคีย์แรก + 128 เช่นเมื่อคีย์ "Q" ถูกกด ค่าสแกนโค้ดแรกจะมีค่าเท่ากับ 16 และเมื่อยกนิ้วขึ้นจากคีย์ ค่าสแกนโค้ดครั้งหลังจะมีค่าเท่ากับ 144 อย่าเพิ่งทำหน้างงกับตัวเลขและวิธีการแปลก ๆ ของคีย์บอร์ดนี้เลยนะครับ ขอให้เข้าใจแต่เพียงว่าคอมพิวเตอร์ไม่ใช่เครื่องมือวิเศษวิโสอะไร มันรู้ว่ากดปุ่มอะไรก็เพราะมีสัญญาณต่อจากปุ่มมาแปลงเป็นตัวเลข ปุ่มนึงก็ตัวเลขนึง แล้วมันจะทราบได้ว่าคุณเลิกกดปุ่มนั้น ๆ แล้วก็เพราะมันส่งสัญญาณที่มีค่าตัวเลขอีกตัวหนึ่ง ซึ่งแสดงว่า "เลิกกดแล้วนะ" มิเช่นนั้นแล้ว ปุ่มที่กดอยู่ก็จะแสดงซ้ำเป็นตัวเดียวกันเรื่อยไป

คีย์บอร์ดกับรอมไบออส (ROM BIOS)

ได้กล่าวไปแล้วข้างต้นว่า ทุก ๆ ขั้นตอนที่เกี่ยวกับคีย์บอร์ดจะถูกส่งไปที่รอมไบออส อะไรล่ะคือ รอมไบออส คุณอาจจะสงสัย ไบออส (BIOS) ย่อมาจาก Basic Input/Output System ซึ่งอยู่ในรอม บริษัทที่ผลิตคอมพิวเตอร์ส่วนมากมักจะปกปิดหวงแหนสิ่งต่าง ๆ ที่บรรจุภายในรอม ในทางตรงกันข้ามไอบีเอ็มเป็นบริษัทที่มีนโยบายแปลกแหวกแนวกว่าบริษัทอื่น ๆ โดยการพิมพ์โปรแกรมและข้อมูลต่าง ๆ ที่อยู่ภายในรอมไบออสเอาไว้อย่างละเอียดยิบในภาคผนวก (Appendix A) ของหนังสือ Technical Reference ซึ่งจัดพิมพ์โดยบริษัทไอบีเอ็มเอง[1] เรื่องของรอมไบออสนี้เป็นเรื่องที่ยืดยาวมากเรื่องหนึ่ง ถ้าพูดมากแทนที่จะเป็นเรื่องคีย์บอร์ด ก็จะต้องเปลี่ยนชื่อเรื่องไปเสียจริง ๆ ในตอนนี้ก็ขอให้ทำความเข้าใจแต่เพียงสั้น ๆ ก่อนว่า รอมไบออสเป็นสิ่งพื้นฐานที่ทำให้คอมพิวเตอร์สามารถเดินเครื่องปฏิบัติงานได้ โดยที่ไบออสจะทำหน้าที่ควบคุมอุปกรณ์ต่าง ๆ ที่ติดต่อเชื่อมโยงกับคอมพิวเตอร์ เช่น มอนิเตอร์, ดิสค์ไดรฟ์, เครื่องพิมพ์ เป็นต้น



รูปที่ 1 แสดงตำแหน่งและสแกนโค้ดของแต่ละคีย์

ที่จริงเรื่องเหล่านี้เป็นเรื่องพื้นฐานที่หลายท่านคงจะรู้ดีอยู่แล้ว แต่ได้เขียนเผื่อไว้สำหรับผู้ที่ไม่คุ้นเคยกับการทำงานของเครื่องสามารถที่จะปะติดปะต่อเข้าใจได้ และบทความนี้จะได้เป็นประโยชน์แก่บุคคลโดยทั่วไปสมเจตนารมณ์ของผู้เขียน สำหรับท่านที่รู้อยู่แล้วก็นึกว่าผมเอามะพร้าวห้าวมาขายสวนก็แล้วกัน ฟังดูแล้วสบายใจดี

ทุก ๆ ครั้งที่คุณกดนิ้วลงบนคีย์และยกนิ้วขึ้น สัญญาณจะถูกส่งไปที่รอมไบออสเป็นอินเตอร์รัพท์ (Interrupt) เบอร์ 9 ซึ่งอินเตอร์รัพท์หรือโปรแกรมย่อยในรอมไบออสเบอร์นี้มีชื่อเรียกอีกอย่างว่า Keyboard Action Interrupt อินเตอร์รัพท์นี้ก็จะเรียก interrupt-handling subroutine ซึ่งจะทำหน้าที่อ่านข้อมูลจากพอร์ทหมายเลข 96 (หรือ hex60) ซึ่งจะได้ค่าสแกนโค้ดออกมา จากนั้น keyboard service routine ก็จะทำหน้าที่แปลงค่าสแกนโค้ดที่ได้นี้เป็นข้อมูลที่มีขนาด 2 ไบท์ โดยมีค่า ASCII อยู่ใน low-order byte และมีค่าสแกนโค้ดอยู่ใน high-order byte ยกเว้นคีย์พิเศษที่ไม่มีค่า ASCII เช่น ฟังก์ชันคีย์ (F1 ถึง F10), คีย์ตัวเลขทางขวาของคีย์บอร์ด คีย์เหล่านี้จะมีค่าของ low-order byte เท่ากับ 0 และมีค่าสแกนโค้ดอยู่ใน high order byte เหมือนปกติ[2] นอกจากนี้ยังมีอินเตอร์รัพท์อีก 3 ตัวที่เกี่ยวข้องกับคีย์บอร์ด คือ 16, 1B, 23[3] แต่ในที่นี้เราจะกล่าวถึงแต่เฉพาะ INT9 หรืออินเตอร์รัพท์ 9 นี้เท่านั้น เพราะ INT9 ทำหน้าที่เกี่ยวกับคีย์บอร์ดในระดับต่ำ คือระดับเริ่มแรกทีเดียว ซึ่งมีผลดีตรงที่ว่าทำให้เราสามารถจัดการกับข้อมูลที่ป้อนเข้าทางคีย์บอร์ดอย่างไรก็ได้ โดยที่ไม่ต้องเข้าไปยุ่งเกี่ยวกับโปรแกรมใด ๆ ที่กำลังรันอยู่เลย ที่จริงเรื่องการทำงานของ INT9 นี้ยังมีอีกยืดยาวเกินขอบเขตของบทความนี้ไป คราวหน้าคราวหลังผมจะเขียนมาเล่าเรื่องโดยเฉพาะเกี่ยวกับอินเตอร์รัพท์ให้ละเอียดครับ ยังมีเรื่องจำเป็นอีกเรื่องหนึ่งก่อนที่เราจะถึงเรื่องโปรแกรม นั่นก็คือเรื่องไบท์แสดงสถานะของคีย์บอร์ด

KB_FLAG at 417H

Bit	Hex	Dec	
7	80	128	Insert mode on
6	40	64	Caps Lock on
5	20	32	Num Lock on
4	10	16	Scroll Lock on
3	08	8	Alt pressed
2	04	2	Left Shift pressed
1	01	1	Right Shift pressed

KB_FLAG_1 at 418H

Bit	Hex	Dec	
7	80	128	Insert pressed
6	40	64	Caps Lock pressed
5	20	32	Num Lock pressed
4	10	16	Scroll Lock pressed
3	08	8	Ctrl-Num Lock (pause key) toggled
2	04	4	PCjr keyboard clicker active
1	02	2	PCjr Ctrl-Alt-caps Lock held
0	01	1	

รูปที่ 2 แสดงสถานะต่าง ๆ ของคีย์บอร์ด

ไบท์แสดงสถานะ (Status bytes)

ไบท์แสดงสถานะ (Status bytes) หรือในบางที่ก็เรียกคีย์บอร์ดแฟล็ก (keyboard flags) ทุกครั้งที่เรากดคีย์ shift ข้างซ้ายรอบไบออสจะเซ็ทบิทที่ 1 ในไบท์ที่ 417H ให้มีค่าเท่ากับ 1 ซึ่งไบท์ที่ 417H นี้เรียกว่าไบท์แสดงสถานะ และยังมีอีก 1 ไบท์คือที่ 418H แต่ละบิทในไบท์ทั้งสองนี้จะแสดงสถานะของคีย์บอร์ดในปัจจุบัน คือมีค่าเท่ากับ 1 เมื่อคีย์ถูกกด และมีค่าเท่ากับ 0 เมื่ออยู่ในสภาพปกติ คีย์ต่าง ๆ ที่แสดงสถานะของคีย์บอร์ดคือ Ins, Caps, Lock, Num Lock, Scroll Lock, Alt, Ctrl, left shift, right shift ซึ่งได้สรุปรวมไว้ในรูปที่ 2

โปรแกรมที่ให้ไว้ได้เขียนเป็นภาษาแอสเซมบลีทั้งหมด สำหรับผู้ที่ไม่คุ้นเคยกับภาษานี้ก็พอที่จะติดตามการทำงานของโปรแกรมได้โดยไม่ยากนัก เพราะผมพยายามที่จะให้คำบรรยายเอาไว้อย่างละเอียดที่สุดเท่าที่จะทำได้ จะใช้เอดิเตอร์ตัวไหนก็ได้แล้วแต่ความถนัด แต่ผมแนะนำให้ใช้ของเทอร์โบปาสคาลเพราะเร็วดี สำหรับแอสเซมเบลอร์ผมสมมติเอาว่าคุณคงใช้ของไอบีเอ็ม คือ มาโครแอสเซมเบลอร์ ส่วนการพิมพ์, แก้ไข และการคอมพายล์ ถ้าเป็นไปได้ควรจะทำในแรมดิสค์สำหรับผู้ที่มีหน่วยความจำมากพอ เพราะจะสะดวกรวดเร็วมากกว่า ในกรณีที่ต้องกลับมาแก้ไขและคอมพายล์ใหม่ซ้ำ ๆ กันหลาย ๆ หน

เพื่อความสะดวกในการคอมพายล์โปรแกรม ควรที่จะสร้างแบตช์ไฟล์ที่ใช้ช่วยประหยัดเวลาและลดข้อผิดพลาดดังนี้

```
ASM %1,,,;
LINK %1,,CON,;
EXE2BIN %1.EXE %1.COM
```

เราใช้ชื่อแบตช์ไฟล์นี้ว่าอย่างไรก็ได้ ในที่นี้ผมสมมติเอาว่าชื่อ BUILD.BAT เมื่อเวลาที่ต้องการคอมพายล์โปรแกรม ก็เรียกชื่อแบตช์ไฟล์นี้ตามด้วยชื่อของโปรแกรม เช่น

```
A > BUILD NOBOOT
```

โปรแกรมตามรูปที่ 3 นี้เป็นโปรแกรมที่ใช้ตรวจสอบการกด Ctrl-Alt-Del ซึ่งเป็นการกดเพื่อ reboot เครื่องคอมพิวเตอร์ โดยที่เมื่อรันโปรแกรมนี้เพียงครั้งเดียว โปรแกรมนี้จะคงอยู่จนกว่าจะปิดเครื่องหรือกดปุ่ม Ctrl-Alt-right Shift ร่วมกับปุ่ม Del หรือ Ins การทำงานของโปรแกรมนี้

รูปที่ 3 โปรแกรม NOBOOT

```
;PROGRAM 1 :  NOBOOT
;
cseg    segment para public 'code'
        org     100h            ;standard code offset for com files
NOBOOT  proc far
        assume cs:cseg,ds:cseg
;
        jump    install         ;do install the new int 9 routine
;
oldint9         dd      0               ;saved original int 9 vector
int9loc         equ     9h*4            ;location of int 9 vector
kb_flag         equ     417h            ;location of keyboard status flag
ctrl            equ     04              ;kb_flag with ctrl bit on
alt             equ     08              ;kb_flag with alt bit on
rshift          equ     01              ;kb_flag with right shift bit on
noctrl          equ     0ffh-ctrl       ;kb_flag with ctrl bit off
delete          equ     83              ;delete keyscan code
insert          equ     82              ;insert keyscan code
;
newint9:                                ;entry here on keyboard interrupt 9
                sti                     ;interrupt enabled
                push    es              ;save requisites
                push    di
                push    ax
                push    cx
                pushf
;
                mov     ax,0
                mov     es,ax           ;es is segment 0
                mov     di,kb_flag      ;kb_flag
                mov     ah,es:[di]      ;into ah
;
                test    ah,alt          ;Alt key down?
                jz      return          ;no,exit
                test    ah,ctrl         ;Ctrl key down?
                jz      return          ;no,exit
;
                in      al,60h          ;get the keypress from 8255
                cmp     al,delete       ;check for Del key pressed
                jne     return          ;no,exit
;
authtest:
                test    ah,rshift       ;right shift down?
                jnz     authuser        ;yes,allow request
;
inhibit:                                ;no,don't allow reboot
                and     ah,noctrl       ;turn off the ctrl flag
                mov     es:[di],ah      ;and put back KB_FLAG
;
authuser:                               ;authorized user requests,
                                        ;let it go through
;
return:
                popf
```

```
                pop     cx
                pop     ax
                pop     di
                pop     es
                jmp     cs:[oldint9] ;to ROM BIOS keystroke processing
;
install:
                mov     ax,0
                mov     es,ax
                mov     di,int9loc      ;save old int 9 vector
                mov     ax,es:[di]      ;int 9 ip
                mov     bx,es:[di+2]    ;int 9 cs
                mov     si,offset oldint9
                mov     [si],ax
                mov     [si+2],bx
;
                mov     ax,0
                mov     es,ax
                mov     bx,ds
;
                cli                     ;disable interrupts
;interrupts disabled
                mov     di,in9loc
                mov     ax,offset newint9
                mov     es:[di],ax      ;change int 9 vector ip
                mov     es:[di+2],bx    ;change int 9 vector cs
;
                sti                     ;reenable interrupts
;
                mov     dx,offset install
                                        ;length of resident portion of program
                int     27h             ;terminate, but stay resident
;
NOBOOT          endp
cseg            ends
end             NOBOOT                                          ■
```

จะอยู่ภายใต้ของโปรแกรมอื่น ๆ ซึ่งยังคงทำงานได้ตามปกติเหมือนไม่มีโปรแกรมนี้ปรากฏอยู่เลย เราเรียกโปรแกรมทำงานที่อยู่ภายใต้โปรแกรมอื่นนี้ว่า Resident Program **ซึ่งมีข้อควรระวังอยู่ประการหนึ่ง** คือไม่ควรจะรันหรือเรียกชื่อโปรแกรมนี้ซ้ำเกินกว่าหนึ่งครั้ง เพราะนอกจากจะไม่มีความจำเป็นที่จะต้องทำเช่นนั้นแล้ว ยังจะก่อให้เกิดผลตามมาที่ไม่คาดคิดด้วย เช่น การแอดเดรสอ้างถึงข้อมูลของโปรแกรมอาจผิดพลาดไปหมด

โปรแกรมชุดนี้ ทำการตรวจสอบการกดปุ่ม Ctrl-Alt-Del ซึ่งมีผลทำให้เครื่องคอมพิวเตอร์รีเซ็ทตัวเองหรือที่เรียกกันว่า reboot หรือ warm boot โดยการเปลี่ยนเวคเตอร์ที่ชี้ไปตำแหน่งที่ INT 9 มาชี้ที่จุดเริ่มต้นของโปรแกรมของเราแทน โปรแกรมจะตรวจสอบการกดคีย์บอร์ด และจะไม่สนใจเมื่อคีย์ Del หรือ Ins ถูกกดพร้อมคีย์ Ctrl-Alt จากนั้นการทำงานก็ถูกส่งต่อไปให้ INT 9 ตามปกติ โปรแกรม NOBOOT นี้เมื่อรันโดยการเรียกชื่อโปรแกรมที่ดอสแล้วก็จะคงอยู่เป็น Resident program จนกว่าจะปิด–เปิดเครื่องใหม่หรือ reboot โดยการกด Ctrl-Alt-right Shift ร่วมกับ Del หรือ Ins

โปรแกรมตามรูปที่ 4 นี้เป็นโปรแกรมสำหรับผู้ที่ต้องการเปลี่ยนคีย์บอร์ดจากแป้นเกษมณีเป็นแป้นปัตตโชติ หลักการใหญ่ ๆ ของโปรแกรมนี้ไม่มีอะไรมาก คือโปรแกรมจะตรวจสอบคีย์บอร์ดว่าอยู่ในโหมดภาษาไทยหรือไม่(สหวิริยา หรือ SIC ใช้คีย์ Scroll Lock สำหรับ toggle ระหว่างภาษาไทยและภาษาอังกฤษ) และอยู่ในแป้นเกษมณีหรือปัตตโชติ ซึ่งโปรแกรมนี้ใช้การกดคีย์ Alt-F2 เพื่อเปลี่ยนระหว่างแป้นทั้งสอง คือ กดครั้งหนึ่งคีย์บอร์ดก็เปลี่ยนจากแป้นเกษมณีเป็นปัตตโชติ และเมื่อกดอีกครั้งหนึ่งคีย์บอร์ดก็เปลี่ยนกลับเป็นแป้นเกษมณีตามเดิม จากนั้นโปรแกรมจะเปลี่ยนรหัส ASCII ในคีย์บอร์ดบัฟเฟอร์มาเป็นรหัสที่ต้องการ แล้วใส่กลับเข้าไปในคีย์บอร์ดบัฟเฟอร์ตามเดิม แต่โปรแกรมนี้เขียนขึ้นเฉพาะผู้ที่ใช้รหัสภาษาไทยของสมาคมมาตรฐานผลิตภัณฑ์อุตสาหกรรมหรือของ SIC เท่านั้น อย่างไรก็ดีผู้ที่ใช้รหัสอื่นก็สามารถที่จะนำไปเป็นแบบอย่างสำหรับของตนได้โดยการแก้ไขข้อมูลที่ออกแบบเอาไว้ในตอนต้นของโปร–แกรมการแก้ไขเพื่อที่จะให้ใช้ได้สำหรับรหัสที่ต้องการ ก่อนอื่นจะต้องทราบเสียก่อนว่ารหัสของตนนั้นบรรจุตัวอักษรต่าง ๆ เอาไว้ที่ใดบ้าง โดยการสั่งให้เครื่องพิมพ์ตัวอักษรพร้อมทั้งรหัส ASCII (โดยมากจะเริ่มตั้งแต่ 161 = “ก”) ที่ใช้บรรจุภาษาไทยออกมา

รูปที่ 4 โปรแกรม **THAIKB**

```
;PROGRAM 2 :    THAIKB
;-------------------------------------------------------------------------------
;
;          CHANGING KESSAMANEE KEYBOARD TO PATTACHOTE KEYBOARD
;
;                              COPYRIGHT 1986
;                                   BY
;
;                          SURAPON PREUGPAIBUL
;
;
;-------------------------------------------------------------------------------
; *available only for FREE distributed, NOT FOR SALE.
; *other using must be given permission by author.
;-------------------------------------------------------------------------------
;
BIOS_SEG        SEGMENT AT 40H                  ;ESTABLISH BIOS SEG
BIOS_SEG        ENDS                            ;ADDRESS.

CSEG    SEGMENT
        ASSUME  CS:CSEG
        ORG     100H                    ;STANDARD CODE OFFSET FOR COM FILES.
BEGIN:          JMP     INITIALIZE      ;GO SET VECTORS, WHEN LOADED BY DOS.
ROM_INT9        DW      ?,?             ;ORIGINAL KEYSTROKE VECTOR ADDRESS
BIOS_KBFLAGS    EQU 17H                 ;BIOS OFFSET TO KEYBOARD SHIFT FLAGS.
BIOS_HEAD       EQU 1AH                 ;BIOS BUFFER ADDRESS OF NEXT OUTPUT KEY
BIOS_TAIL       EQU 1CH                 ;BIOS BUFFER ADDRESS FOR NEXT INPUT KEY
ALT             EQU 08                  ;BITS SET WHEN ALT KEY PRESSED
SCRLL           EQU 10H                 ;BITS SET WHEN IN SCROLL LOCK STATE
F2              EQU 60                  ;SCAN CODE, WHEN F2 IS PRESSED
INDEX             EQU 160                 ;INDEX TO ADDRESS DATA IN TRANSLATE TABLE
PAT_FLAG        DB       0              ;1,IF IN PATTACHOT KEYBOARD
;-------------------------------------------------------------------------------
;               TRANSLATE TABLE
;
PAT_TABLE       DB      0,167,49,0,56,0,184,162
                DB      48,176,54,166,236
                DB      178,230,211,207,173
                DB      191,171,161,57,53
                DB      202,201,199,227,187
                DB      186,190,205,233,52
                DB      208,225,193,235,172
                DB      204,228,226,170,224
                DB      183,169,203,192,182
                DB      195,232,185,194,212
                DB      180,55,164,217,216
                DB      147,0,0,0,0
                DB      0,209,197,179,168
                DB      181,0,231,215,210
                DB      213,189,188,200,214
                DB      0,0,234,34,37
                DB      44,63,46,40,41
                DB      45,43
;-------------------------------------------------------------------------------
;               SPECIAL CHARACTERS TRANSLATE TABLE
;
SPECIAL_CHAR    EQU     13                  ;NUMBER OF SPECIAL CHARACTERS
TRANSLATE_OLD   DB      255,34,40,41,46,63,38,95,47,35,66,160,44
TRANSLATE_NEW   DB      255,196,174,175,177,206,95,51,50,33,147,61,198
;*******************************************************************************
INT9:                                   ;ENTRY OF KEYBOARD INT9 CALL
;*******************************************************************************
        PUSH    DS                      ;SAVE CALLER'S DATA SEGMENT ADDRESS
        PUSH    BX                      ;SAVE REQUISITES
        PUSH    SI
        PUSH    AX
        PUSH    CX
        MOV     BX,BIOS_SEG             ;SET UP ADDRESSING TO BIOS DATA SEGMENT
        MOV     DS,BX
        ASSUME  DS:BIOS_SEG
        MOV     BX,DS:[BIOS_TAIL]       ;SAVE CURRENT KEYBOARD TAIL LOCATION
        PUSHF                           ;CALL ROM BIOS KEYSTROKE PROCESSING
```

```
        CALL    DWORD PTR ROM_INT9      ;INTERUPT, AND RETURN HERE WHEN DONE
        CMP     BX,DS:[BIOS_TAIL]       ;IF NEW CHARACTER ENTERED, TAIL MOVED
        JNE     NEW_KEY
        JMP     ALL_DONE                ;NO, RETURN TO USER
;
NEW_KEY:
;
        TEST    BYTE PTR DS:[BIOS_KBFLAGS],ALT ;SEE IF ALT IS PRESSED
        JZ      RET_FLAG
        MOV     AX,[BX]
        TEST    AH,F2                   ;SEE IF F2 IS PRESSED
        JE      SET_FLAG
        TEST    BYTE PTR PAT_FLAG,1     ;SEE IF IN PATACHOT KEYBOARD?
        JZ      SET_FLAG                ;NO, GO SET FLAG
        MOV     PAT_FLAG,0              ;IF IN PATACHOT, TOGGLE FLAG OFF
        JMP     RET_FLAG
;
SET_FLAG:
        MOV     PAT_FLAG,1
RET_FLAG:
C       TEST    BYTE PTR DS:[BIOS_KBFLAGS],SCRLL ;SEE IF SCROLL LOCK ON
        JZ      ALL_DONE                ;NO, WE ARE IN ENGLISH MODE
        TEST    BYTE PTR PAT_FLAG,1     ;SEE IF IN KESSAMANEE KEYBOARD
        JZ      ALL_DONE
        CMP     BYTE PTR [BX],INDEX     ;SEE IF CHARACTER ABOVE 160?
        JA      NORMAL                  ;THEN IT IS NORMAL CHARACTER
        MOV     CX,SPECIAL_CHAR         ;HANDLE SPECIAL CHARACTERS
START_LOOP:
        MOV     SI,CX
        MOV     AL,[BX]
        CMP     AL,TRANSLATE_OLD[SI]
        JE      FOUND
        LOOP    START_LOOP              ;LOOP UNTIL  ALL SPECIAL  CHARS ARE CHECKED

        JMP     ALL_DONE
FOUND:  MOV     AL,TRANSLATE_NEW[SI]
        JMP     PUT_IN
NORMAL:
        MOV     SI,[BX]                 ;GET CHARACTER ASCII
        SUB     SI,INDEX                ;FIND POINTER BY MINUS 160
        AND     SI,00FFH                ;MASK OFF HIGH BYTE
        MOV     AL,PAT_TABLE[SI]        ;GET VALUE STORE IN AL
PUT_IN:
        MOV     [BX],AL                 ;THEN INTO KEYBOARD BUFFER POINTED BY BX
;
ALL_DONE:                               ;ALL DONE.  KEY HAS BEEN PROCESSED BY
        POP     CX                      ;ROM BIOS
        POP     AX
        POP     SI
        POP     BX
        POP     DS
        IRET                            ;RETURN TO USER.
;
;*****************************************************************************
INITIALIZE:                     ;INITIALIZE INTERRUPT VECTORS 8 AND 9
;*****************************************************************************
        ASSUME  DS:CSEG
        XOR     AX,AX
        MOV     ES,AX                   ;SET ES TO ABSOLUTE 0 (START OF VECTORS)
        MOV       CX,ES:[9*4]             ; CX=OFFSET ADDRESS TO ROM KEYSTROKE CODE

        MOV     DX,ES:[9*4+2]           ; DX=SEGMENT  ADDRESS OF  ROM KEYSTROKE CODE

        MOV     ROM_INT9,CX
        MOV     ROM_INT9[2],DX
        CLI
        LEA     AX,INT9                 ;AX=OFFSET TO OUR KEYSTROKE CODE
        MOV     ES:[9*4],AX             ;RESET KEYSTROKE VECTOR TO POINT TO
        MOV     ES:[9*4+2],CS           ;OUR OFFSET AND CODE SEGMENT
        STI
        LEA     DX,INITIALIZE           ;POINT TO END OF CODE WE NEED TO KEEP
        INT     27H                     ;AND TERMINATE SETUP OPERATION
CSEG    ENDS
        END     BEGIN
```

ทุกตัว จะใช้ภาษาอะไรก็ได้ตามถนัด เช่น ตัวอย่างข้างล่างเป็นภาษาเบสิกง่าย ๆ ดังนี้

```
100 FOR I = 33 TO 255
110 PRINT I; " "; CHR$(I),
120 NEXT
```

การที่ให้เริ่มตั้งแต่ 33 ก็เพราะอาจจะมีการโยกย้ายสลับที่ของตัวอักษรที่มีค่าต่ำกว่าตัวอักษรภาษาอังกฤษ เช่น ในภาษาไทยก็มีเครื่องหมาย " ใช้รหัส ASCII ของ " เดียวกับภาษาอังกฤษคือ 34 เป็นต้น การกดแป้นที่เป็นตัว " บนแป้นเกษมณี (คือคีย์ W) แล้วให้กลายมาเป็นตัว ฤ บนแป้นปัตตโชติได้ เราจะต้องทราบรหัส ASCII ทั้งของตัว " และตัว ฤ จากนั้นก็ให้เทียบเคียงกับรหัสของ SIC ว่าตัวใดอยู่ที่ใด แล้วให้เปลี่ยนกับค่าในข้อมูลในโปรแกรม เช่นถ้ารหัสตัว โ ของคุณเท่ากับ 210 ส่วนของ SIC มีค่าเท่ากับ 226 คีย์ตัว โ ในแป้นเกษมณีจะต้องพิมพ์ออกมาเป็นตัว ณ ในแป้นปัตตโชติ และถ้ารหัสตัว ณ ของคุณเท่ากับ 177 ดังนั้นที่ตำแหน่งข้อมูลที่ 226 จะต้องมีค่าเท่ากับ 177 ในโปรแกรมนี้ใช้ index เริ่มที่ "ก" คือเริ่มที่ค่า 161 ดังนั้นตำแหน่งที่ต้องแก้คือที่ 226–161 = 65 ถ้าดูในโปรแกรมที่เป็น normal character หรือข้อมูลใน PAT TABLE ข้อมูลที่ 65 จะมีค่าเท่ากับ 179 ซึ่งก็คือรหัสตัว "ณ" ของ SIC นั่นเอง (รูปที่ 5)

แต่มีตัวอักษรพิเศษบางตัวที่ต้องตรวจสอบต่างหาก โดยเฉพาะตัวอักษรที่มีรหัส ASCII เดียวกันกับภาษาอังกฤษ เช่น ตัว " , . = – เป็นต้น มักจะโดนย้ายไปอยู่ในที่ต่าง ๆ กัน ทั้งบนแป้นเกษมณีและแป้นปัตตโชติ คือถึงแม้จะใช้รหัสเดียวกันกับภาษาอังกฤษแต่อยู่คนละคีย์กัน ส่วนของตัวอักษรเหล่านี้ในรหัสของ SIC มีทั้งหมด 12 ตัว เรามีทางเลือกอยู่ไม่มากที่จะจัดการกับการเปลี่ยนแปลงคีย์ เช่น โดยการสร้างตารางใหญ่เพื่อใช้บรรจุรหัสทั้งหมดตั้งแต่ค่า 33 จนถึงประมาณ 254 หรือโดยการแยกจัดการกับตัวอักษรที่ไปแทนที่คีย์ที่มีค่า ASCII ต่ำกว่าตัว ก คือต่ำกว่า 161 ดังเช่นตัว ฤ (รหัส 196) ต้องเข้าไปแทนที่ " (รหัส 34) (ให้กดคีย์ " บนแป้นเกษมณีแล้วออกมาเป็นตัว ฤ บนแป้นปัตตโชติได้จะต้องแทนที่รหัส 34 นี้ด้วยรหัส 196 ซึ่งเป็นตัว ฤ นั่นเอง) ซึ่งในโปรแกรมนี้จะแยกจัดการกับตัวอักษรเหล่านี้ซึ่งมีจำนวนไม่กี่ตัว ดีกว่าไปเปลืองเนื้อที่ในการสร้างตารางใหญ่ขึ้นมาอีกอันหนึ่ง ตัวอักษรที่ถูกจัดการเป็นพิเศษนี้อยู่ใน TRANSLATE OLD จะถูกเปลี่ยนมาเป็น TRANSLATE NEW ในแป้นปัตตโชติ

33	!	34	"	35	#	36	$	37	%	38	&
39	'	40	(	41	)	42	*	43	+	44	,
45	–	46	.	47	/	48	0	49	1	50	2
51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>
63	?	64	@	65	A	66	B	67	C	68	D
69	E	70	F	71	G	72	H	73	I	74	J
75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V
87	W	88	X	89	Y	90	Z	91	[	92	\
93	]	94	^	95	_	96	'	97	a	98	b
99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n
111	o	112	p	113	q	114	r	115	s	116	t
117	u	118	v	119	w	120	x	121	y	122	z
123	{	124	¦	125	}	126	~	127		128	┌
129	┐	130	└	131	┘	132	|	133	–	134	├
135	┤	136	┴	137	┬	138	┼	139	■	140	←
141	↑	142	→	143	↓	144		145		146	∴
147	⌘	148	⌘	149	∴	150	∴	151	⌘	152	⌘
153	⌘	154	⌘	155	⌟	156	⌘	157	⌘	158	⌟
159	เ	160		161	ก	162	ข	163	ข	164	ค
165	ค	166	ฆ	167	ง	168	จ	169	ฉ	170	ช
171	ช	172	ฌ	173	ญ	174	ฎ	175	ฏ	176	ฐ
177	ฑ	178	ฒ	179	ณ	180	ต	181	ต	182	ถ
183	ท	184	ธ	185	น	186	บ	187	ป	188	ผ
189	ฝ	190	พ	191	ฟ	192	ภ	193	ม	194	ย
195	ร	196	ฤ	197	ล	198	ฦ	199	ว	200	ศ
201	ษ	202	ส	203	ห	204	ฬ	205	อ	206	ฮ
207	ฯ	208	ะ	209	◡	210	า	211	ำ	212	◢
213	◢	214	◢	215	◢	216	'	217	"	218	'
219	◢	220	⌘	221	⌘	222	◢	223	●	224	เ
225	แ	226	◢	227	◢	228	◢	229	า	230	ๆ
231	◢	232	.	233	◡	234	◢	235	.	236	◢
237	.	238	◢	239	●	240	๐	241	๑	242	๒
243	๓	244	๔	245	๕	246	๖	247	๗	248	๘
249	๙	250	๚	251	◢	252	⌘	253	⌘	254	◢
255											

รูปที่ 5 รหัสมาตรฐานของ SIC

สรุป

การเปลี่ยนค่าข้อมูลที่อยู่ในคีย์บอร์ดบัฟเฟอร์เป็นข้อมูลอะไรก็ได้ นอกจากการเปลี่ยนตัวหนังสือตามแป้นเกษมณีหรือปัตตโชติ สามารถที่จะทำได้โดยใช้หลักการเดียวกัน คือใช้ resident program และเปลี่ยนเวคเตอร์ที่ใช้ชี้ไปที่ INT9 มาเป็นจุดเริ่มต้นโปรแกรมของเรา ภายในโปรแกรมของเรานี้จะเรียกไปที่ INT9 อีกต่อหนึ่ง สามารถกล่าวอีกอย่างหนึ่งได้ว่า เราใช้โปรแกรมของเราเข้าไปคั่นระหว่างการทำงานของโปรแกรมอื่นกับรอมไบออสนั่นเอง สำหรับผู้ใดที่ยังข้องใจหรือสงสัยประการใด ถามมายังผมโดยผ่านทางหนังสือไมโครคอมพิวเตอร์ได้ครับ

เชิงอรรถ

1) **IBM Personal Computer XT Technical Reference manual, Revised Edition, April 1983.**

2) รายละเอียดของ **interrupt 9** นี้สามารถหาอ่านได้ใน **IBM Technical Reference** หน้า **A–28** ตรงที่เขียนเลเบิลไว้ว่า **KB INT**

3) ผู้ที่สนใจสามารถศึกษาเกี่ยวกับ **INT 6** และ **resident program** ได้ในไมโครคอมพิวเตอร์ ฉบับที่ 21 เดือนมีนาคม 2529 เรื่อง "สร้างโปรแกรมไว้ใต้โปรแกรม" ของคุณบุญเลิศ เอี่ยมทัศนา หน้า 121